NASA CR-
141477

(NASA-CR-141477) EARTH RESOURCES DATA ANALYSIS PROGRAM, PHASE 2 Final Report, Jun. 1973 - May 1974 (Rice Univ.) 38 p HC $3.75 CSCL 09B

N75-15333

Unclas
G3/61 06665

ICSA

INSTITUTE FOR COMPUTER SERVICES AND APPLICATIONS

RICE UNIVERSITY

Phase II of the Rice University Earth Resources Data Analysis Program

FINAL REPORT
(June, 1973 - May, 1974)

Institute for Computer Services and Applications
Rice University
Houston, Texas 77001

June, 1974

This research was sponsored by NASA under Contract NAS 9-12776

I. INTRODUCTION

During the preceding contract year, three tasks have been undertaken. They are:

1) Applications Development System (ADS) Analysis,

2) Algorithmic Development,

and 3) Evaluation of Technical Reports.

Task 1 (see Section II) consisted of a detailed study of the needs of EOD with respect to an applications development system (ADS) for the analysis of remotely sensed data; followed by an evaluation of four existing systems (ERIPS, ASTEP, LARSYS batch, and LARSYS 3) with respect to these needs; and concluded with a set of recommendations as to possible courses for EOD to follow to obtain a viable ADS. Task 2 (see Section III) comprised several subtasks of which three were continuations of projects initiated during our first year's contract. These include two algorithms for multivariate density estimation, a data smoothing algorithm, a method for optimally estimating prior probabilities of unclassified data, further applications of the modified Cholesky decomposition in various calculations, and a few other projects. Little effort was expended on task 3 (see Section IV) due to a shift in priorities mostly necessitated by the increased effort devoted to task 1. However, two reports were reviewed.

This report summarizes both the efforts and the findings of the above project. Each of the tasks are described in the following sections.

II. TASK 1: Applications Development System Analysis

This study (see the Task 1 Final Report) describes the results of a detailed study of the needs of EOD for an applications development system (ADS), including a detailed evaluation of four existing systems (ERIPS, ASTEP, LARSYS batch, and LARSYS 3) with respect to these needs. Suggested courses of action are proposed for the EOD to pursue.

The original task definition in the contract called for:

1) Developing a set of design goals for an applications development system (ADS),
2) Evaluating ASTEP and the LARSYS batch programs to determine whether either met these goals,
3) Recommending courses of action for the future of these systems:
    a) If neither system meets the design goals, develop a system design for an ADS that does;
    b) Determine the mutual impact of this system on either the IBM 360/75 under RTOS or the UNIVAC 1110 under EXEC 8;
    c) Develop a recommended approach to the development of a data analysis ADS at JSC,
4) Determining whether a terminal to ASTEP, a LARS terminal, or set of batch programs is the most desirable method of transporting remote sensing ADP technology to an agency establishing a program in remote sensing.

This task definition was later modified to include ERIPS and LARSYS 3 in the evaluation study and to exclude item 4 above.

It is important to remark here that some specific constraints of the EOD's were not taken into account in this study. Such considerations not involved in any quantitative fashion in this analysis include:

1) cost of implementing recommended modifications,

2) system performance and response as a function of number of users,

3) availability and capacity of hardware,

and 4) specific hardware implementations.

The method employed for conducting this study was to adopt a "top-down" approach to the evaluation process. This consisted of:

1) Developing design goals for an ideal ADS. These goals represent general areas of interest that such a system must address, and are not of themselves prioritizable.

2) Detailing supporting design objectives for those design goals. These objectives are specfic functional capabilities that an ideal ADS should have.

3) Prioritizing these design objectives with respect to the needs of EOD.

4) Rating the various systems on each of the design objectives to indicate how well each satisfied the requirements of each objective.

5) Recommending alternative courses of action for the EOD to follow based on the above findings.

Before discussing further the methodology employed and the results of this task, it is important to establish a framework for the ideal ADS. Such a system will be used to develop and test new algorithms and procedures for various remote sensing applications. Its basic characteristics should include that it be easy to use for a wide variety of personnel, accessible and responsive to users, reliable, and as flexible and complete a system as possible. The system must serve two kinds of users - production and techniques development personnel. The production user needs to be able to efficiently process large amounts of data using state-of-the-art techniques. He requires that results be in form suitable for presentation or further analysis. The techniques development person, on the other hand, needs a system where he can thoroughly test and evaluate new algorithms and techniques. The system thus should be easily modifiable and require the user to have only a minimum of knowledge of the internals of the system. The user should be able to easily add, delete, replace, or modify any of the algorithms in use for his own purposes, while assuring the integrity of the standard system.

The design goals were established to present general areas that a TDS should address. These areas are, in summary:

i) Combination of production and test systems in a unified framework,

ii) Simplification techniques for system maintenance and enhancement,

iii) Data and system management facilities,

iv) Graceful degradation features,

v) Convenience features,

vi) System measurement and evaluation features,

vii) Basic system analysis functions.

These goals as such are not prioritizable since they do not represent specific functional capabilities. The design objectives, on the other hand, are prioritizable. They are specific capabilities for an ideal ADS to have. These objectives were then prioritized according to the needs of the EOD program objectives. Ratings of from one to four were assigned where the ratings are:

Priority	Description
1	Necessary to achieve EOD program objectives
2	Necessary to achieve a high level of EOD's program objectives
3	Desirable feature
4	Questionable desirability

The various systems were then rated on each of the design objectives. Ratings were assigned from zero to five in the basis of how well each system functionally met each objective. The rating codes and their meanings are:

Rating	Meaning
5	Exceeds requirements of this objective
4	Meets all requirements of this objective
3	Satisfies most of the requirements of this objective
2	Satisfies some of the requirements of this objective
1	Satisfies only a small portion of the requirements of this objective
0	Does not have any such capability as specified by this objective.

The results of this evaluation process comprise the bulk of the Task 1 final report. These findings are briefly summarized below.

ERIPS possesses several key features, notably an extensive, interactive imaging capability and an abundance of user convenience features. Though ERIPS is highly modular, it was not designed for modification by the user community: most of the coding is in a specially designed assembler language and the programming skill necessary to understand the internals of the system are far beyond the average user.

ASTEP, on the other hand, was written mostly in FORTRAN V and the coding is relatively easy to decipher. However, no modification aids for the user are available and documentation is not very extensive. Though ASTEP can be run in an interactive mode, the use of tapes is limited by operational difficulties and, thus, system use is limited. Additionally, no interactive imaging capabilities exist.

The LARSYS batch programs were also written mostly in FORTRAN V. However, very little documentation exists on these programs, thus making modification a difficult chore. The most serious problem with these programs is that though many functions are available, they are not in a unified system, which creates a myriad of problems for users and programmers alike. The lack of interactive and interactive imaging capabilities further hampers the utility of these programs.

LARSYS 3 possesses many of the essential features of the ideal ADS. It too is written mostly in FORTRAN, and extensive documentation is readily available. A variety of modification aids eases somewhat the user's task, but other such features do not currently exist. The system is relatively easy to use, has several modes of operation, and a training program is available. An interactive, imaging device exists at Purdue, but none are supported elsewhere. It is presently lacking in basic systems analysis functions available, but the structure exists for later adding these.

Thus, in terms of which system comes closest to meeting the requirements for an ADS, LARSYS 3 appears to be the most suitable in principle. If LARSYS 3 is to be used most effectively as an ADS, it is worth examining what modifications are necessary to further enhance its utility, and how difficult would such modifications be to make. This study would indicate that modifying LARSYS 3 in several specific areas would produce an ADS which would satisfy most of the needs of EOD. The major areas of modification include adding more analysis functions, adding a more extensive imaging capability, improving the modifiability characteristics of the system, probably converting the system to run under the IBM Time Sharing Option (TSO), and installing it on IBM 370/158 or 168. These latter two modifications are to allow EOD to have their own system locally with mainline IBM support and file compatibility with other IBM computers (because of using TSO rather than CMS). Compared with operating remotely from Purdue, this would eliminate difficult problems of supporting remote interactive imaging devices, transferring bulk data over long distances, configuration control and future growth of the system, and overloading the system at Purdue. This may well represent the best courses of action for EOD in terms of capabilities for satisfying their needs for an ADS.

If the above is not possible, one alternative method would be to provide an interactive image display tied into the LARSYS 3 system at Purdue. This would require intelligent (perhaps specially designed) terminals to effectively provide this ability over the long distances involved,and high bandwidth communication lines. Other modifications to the system as suggested above could be made to LARSYS to increase its utility. However, difficulties may be encountered in the areas of overloading the system and transportation of data back and forth. Such a configuration would have a substantially lower throughput and turnaround capacity, but may be suitable for relatively low volume demand.

Two other possiblities for an interactive ADS suggest themselves: build an entirely new system based on the ideal design goals and objectives contained in the Task 1 report, or radically modify the internals of ERIPS. Developing a new system based on the established design objectives would be a very costly project both in time and money, but it would probably provide a very effective means of doing techniques development work. Modifications necessary to effectively utilize ERIPS as an ADS consist of establishing terminals in Building 17 and providing users with the capability to work with the internals of the system. The latter would entail re-programming all algorithmic routines into high level language; providing interfaces to other system routines which would allow users to perform such tasks as menu generation using only the high level language; and adding numerous other capabilities to the system. We do not highly recommend this approach since it appears that a relatively large amount of effort must be expended, and the resulting system would still not be entirely satisfactory from the modifiability standpoint.

Modification of either ASTEP or LARSYS batch is not recommended. The basic structures of both of these would not be able to accomodate the necessary modifications. However, parts of these systems, particularly some of the algorithms, could be used with minor modifications in developing a new system or as additional functions in LARSYS 3.

III. TASK 2: Algorithmic Development

Non-parametric density estimation:

Two methods of attacking this problem are being investigated. The aim of these projects is to provide a computationally viable way of estimating multivariate density functions from relatively small sample sizes, and then to devise a classifier using this model rather than the standard Gaussian one. Such a classifier could significantly increase classification accuracy and also enable one to avoid splitting and later recombining multimodal classes.

The present efforts are directed towards estimating the densities with little concern being given to computational efficiency. It is felt that once such algorithms can be effectively used, methods for greatly increasing their efficiency will be developed or special purpose hardware could be designed.

a) This algorithm (see the "Estimation of Multivariate Probability Density Functions Using B-Splines" by J. O. Bennett) estimates a p-dimensional density function given n random p-vectors of data. The data is first transformed to make it pseudo-independent (the covariance matrices are transformed into identity matrix). Then a p-dimensional density kernel estimator is used with a p-fold tensor product of B-splines as basis functions. The estimator is proven to be consistent in the integrated mean square error sense.

This method developed from an earlier algorithm - spline smoothing of histograms. The difficulty with the previous method was that in many dimensions, histogramming becomes an arduous task because of the number of bins involved. Thus, though this algorithm functioned quite well in one dimension, the results were not readily extendable to the multi-dimensional case. The new algorithm avoids the histogramming problem entirely.

All of this leads to an algorithm which yields a "good" estimate of a multivariate density function even with small sample sizes of training data. This algorithm has been implemented and tested using random numbers from a variety of distributions. Performance has been quite satisfactory. It has also been installed in the version of LARSYS operating on Rice University's IBM 370/155 to compare it with Gaussian maximum likelihood classifier. Results of these tests show that, though the B-spline estimator is relatively slow, its performance on "Gaussian-like" data is comparable to a Gaussian estimator; whereas on other distributions (e.g., bimodal), its performance is significantly better.

The algorithm as presently implemented in the classification section of LARSYS is slow. This is mainly due to the fact that the estimate of the value of the density function of a class for an arbitrary data point involves (1) a rotation of the data vector, and (2) the calculation of the value of a cubic B-spline for each dimension and each data vector from the training samples. For large training sample sizes, the second of these features can entail a very large amount of computation. However, a few points suggest ways for alleviating this problem. First of all, cubic B-splines have finite support and thus need not always be explicitly evaluated. Also, schemes for ordering the training data can be used to avoid performing many of the computations. In addition, for many applications, one can use linear basis functions instead of cubic, thus considerably reducing the number of computations necessary.

At this point, we can suggest guarded optimism for the applicability and usefulness of this algorithm in remote sensing applications. More testing with remote sensing data needs to be done to determine how generally successful the algorithm is in the environment. Improvements to this algorithm are currently being investigated.

b) In this study we consider the problem of estimating the probability functions $\omega \in L^1[a,b]$ which gave rise to the random samples $\{x_1, x_2, \dots, x_n\}$. The interval $[a,b]$ may be either infinite or finite.

Recall that by $L(v)$, the likelihood that $v \in L^1[a,b]$ gave rise to the samples $\{x_1, x_2, \dots, x_n\}$, we mean

$$L(v) = \prod_{i=1}^{N} v(x_i)$$

Let S be a manifold in $L^1[a,b]$. By the maximum likelihood estimate corresponding to the samples $\{x_1, \dots, x_n\}$ and the manifold S, we mean the solution of the following optimization problem:

$$\text{maximize } L(v); \quad \text{subject to}$$

$$v \in S, \; v(t) \geq 0 \;\; \forall\, t \in [a,b] \quad \text{and} \quad \int_a^b v(t)\, dt = 1$$

It is well-known that the parametric likelihood estimate (S is finite dimensional) is well defined. However, a finite dimensional manifold does not approximate well. Hence it makes sense to consider nonparametric maximum likelihood estimation (infinite dimensional S). Clearly, if the manifold S can approximate the Dirac delta function, i.e., contains nonnegative functions whose support is a given small interval centered at $x \in [a,b]$, integrate to 1 and have arbitrarily large values at x, then our optimization problem has no solution. Moreover, this approximation property is enjoyed by most infinite dimensional manifolds in $L^1[a,b]$; hence, we should not expect the nonparametric maximum likelihood estimation problem to have a solution. The situation in present-day applications is actually worse, for it is often the case that in the parametric case we choose S from a seouence of manifolds $\{S_m\}$ where the dimension of S_m is m, $S_m \subset S_{m+1}$ and $\bigcup_{m=1}^{\infty} S_m$ is dense in $L^1[a,b]$; hence the problem is definitely unstable and somewhat ill-defined. Namely we are motivated to choose m large so that we can better approximate the probability density giving rise to the samples; however, for a large m the problem approximates a problem which has _no_ solution.

The previous remarks motivated the maximum penalized likelihood estimate; which consists of replacing the functional L in our optimization problem with the functional $\hat{L}$ defined by

$$\hat{L}(v) = \prod_{i=1}^{N} v(x_i) \quad \exp(-||v||^2)$$

where the norm $||\cdot||$ is some appropriate norm on the manifold S. We consider many interesting maximum penalized likelihood estimators and show that they are well defined. We also show that some maximum penalized likelihood estimators are splines and give some numerical examples. (See the technical report "Nonparametric Maximum Likelihood Estimation of Probability Densities by Penalty Function Methods" by G. F. de Montricher, R. A. Tapia, and J. R. Thompson).

Use of Spatial Information:

Using interpolation polynomials of odd degree, a method of detecting and correcting errors in equally spaced data is presented. This method permits one point to be corrected without contaminating good points. To each point, the method associates an error that measures the distance between this point and the polynomial that interpolates certain neighboring points. By selecting the points with the largest error and moving it so that the error decreases, a smoother set of data is produced. The method is said to be local because if just one point is bad, it is going to be detected and corrected without disturbing its neighbors. By successively selecting the points with the greatest error and modifying them, smoother data is produced. To measure smoothness and to give another interpretation to the error of point, we use the fact that the distance between a point and the polynomial of degree 2k-1 that interpolates its 2k neighbors is the 2k-th divided difference. Therefore, by smoothness, we understand the summation of the squares of the k-th divided differences of each point, and by moving the point with the greatest 2k-th divided difference, the smoothness will decrease the most. It is proved that ultimately the method converges, therefore the possibility of moving one point back and forth is excluded.

By preprocessing some of the data, a threshold error can be found beyond which data is said to be smooth. This error is given in terms of the ratio between the average of all the point errors of the original data and maximum point error in each iteration. Another interesting feature of this method is that it simulates the fairing or smoothing of data points as performed by a human. In such cases, the method will smooth the data until the error associated with each point is impossible to detect by human sight. The value of this minimum error is furnished by modern psychology.

This method, when applied to C-1 flight line data, improved the performance of classification in each of the different classes considered by approximately 5 percentage points. Here each channel and line of pixels were independently smoothed, but the method is readily extendable to smoothing in both physical dimensions.

It was then compared with spline smoothing. In addition to being considerably faster, our method yields more consistent improvement in accuracy whereas spline smoothing sometimes oversmoothes the data (see the report "Error Detection and Data Smoothing Based on Local Procedures" by V. M. Guerra).

A new approach to the problem of estimating proportions:
The problem of estimating acreages or proportions of the several crops under consideration has received attention recently in the research literature. In some acreage estimation problems, it is realistic to assume that the signatures of the several classes have well-known statistics, while their proportions are unknown. In estimating the acreage of wheat, for example, we can model the problem as a two class case, where in class 1 we have wheat, and in class 2 everything else. The basic difficulty in estimating acreages lies in the fact that the estimate must be based on unclassified noisy data. If the data are classified with zero error probability, the problem is trivial, and a simple "relative frequency" estimate is intuitively and theoretically satisfying. In order to have Bayes classification rule, knowledge of the prior probabilities (or proportions) is necessary. On the other hand, in order to estimate the prior probabilities we need to classify a sufficient amount of data. Hence, in order to have a decent performance in classification and estimation of priors, it is profitable to look at the coupled problem of Bayes classification and simultaneous estimation of prior probabilities.

A report has been published on this subject, with the title "Optimal Design with Unknown Priors" by D. Kazakos. In this report, a sequential scheme for simultaneous classification of data and updating the estimated prior probabilities is proposed and analyzed. The probability density functions under each of the M classes are assumed known, and the prior probabilities are assumed unknown and sequentially estimated. It is proved that the scheme converges to the true value of the prior probabilities, and hence the adaptive classification scheme converges to the Bayes classifier. Furthermore, a significant property of the scheme is that the error variance of the estimate of the prior probabilities converges to zero as N^{-1}, where N = number of observations. This is significant, because even if we had a set of N perfectly classified observations, the error variance of the relative frequency estimate would converge to zero as N^{-1}. The recursive form of the estimation scheme makes it attractive for situations where the proportions are varying. The method can "track" slowly varying proportion vectors.

Other variations of the proposed method are currently under investigation. In a forthcoming report, theoretical and numerical comparisons of several related proportion estimation methods will be presented.

Numerical Optimization of Algorithms:

This task is concerned with developing numerically optimal algorithms for use in remote sensing analysis. It is our view that the algorithms employed in remote sensing applications be as accurate as possible since use of unreliable algorithms can lead to inaccurate results, and, possibly, very erroneous interpretations. Such difficulties might be very hard to detect when they occur and could cause considerable delays. Also, the algorithms used should be as efficient as possible to conserve computer time and thus the resources of the project. In the past, we have shown how the modified Cholesky decomposition (MCD) may be employed in many computations where the covariance matrices and their inverses are used. This has effected a considerable savings in computation time and increased accuracy over algorithms previously in use. Memorandum on the following applications of the MCD are included in this report:

1) Computing the average weighted divergence

2) Computing the interclass divergence (for use in calculating the transformed divergence)

3) Performing feature selection using D. Tebbe's criterion.

Feature Selection:

Another minor project in this task is to devise an algorithm for performing feature selection (or extraction) using the condition numbers of the covariance matrices as a measure of the separability of the classes. The condition number of a matrix is defined by

$$\text{cond}\ (A) = ||A|| \quad ||A^{-1}||$$

where larger condition numbers represent matrices whose rows (columns) are more nearly linearly dependent. Then the rationale for applying this measure to the feature selection problem is to find the subspaces containing most of the information and thus to have the rows (columns) of the covariance matrices be as orthogonal as possible.

A program was written to select subsets of channels from C1 data and to pick a subset of a specified size that minimized the maximum condition number of the covariance matrices. Classification was then run using subsets selected by the average divergence criterion (a d c). Performance decreased relative to using channels selected by the a d c. As a next step, we hope to examine how the condition numbers vary as the a d c selects subsets, hopefully to gain insight into what are suitable criteria to employ when using condition numbers as a distance measure.

IV. TASK 3: Review of Technical Reports

G. Austin's memos "Analysis of LARS Subroutine CLASS and Recommended Coding Improvements to Reduce Its Execution Time" and "Modifications to ERIPS Requirements" have been reviewed. A change in priorities prevented other reports from being reviewed.

APPENDICES

RICE UNIVERSITY

Institute for Computer Services
and Applications

MEMORANDUM

DATE: Aug. 15, 1973

TO: K. Baker

FROM: D. L. Van Rooy

RE: Use of the Modified Cholesky Decomposition in Interclass Divergence Calculation.

To compute the transformed divergence, the interclass divergence $D(i, j)$ is needed. An efficient and numerically stable method for computing the $D(i, j)$ is to employ the modified Cholesky decomposition of the covariance matrices. This note will derive the appropriate expressions for doing this.

I. Derivation

The interclass divergence $D(i, j)$ is given by

$$D(i,j) = D_1(i,j) + D_2(i,j) \tag{1.1}$$

with

$$D_1(i,j) = \tfrac{1}{2}\, \mathrm{tr}\, \left[(K_i - K_j)(K_j^{-1} - K_i^{-1}) \right] \tag{1.2}$$

and

$$D_2(i,j) = \tfrac{1}{2}\, \mathrm{tr}\, \left[(K_i^{-1} + K_j^{-1})(\mu_i - \mu_j)(\mu_i - \mu_j)^* \right] \tag{1.3}$$

where K_i is the i^{th} covariance matrix ; μ_i , the corresponding mean victor, and tr denotes the trace. D_1 can now be simplified to

$$D_1 = \tfrac{1}{2}\, \mathrm{tr}\, (K_i K_j^{-1}) + \tfrac{1}{2}\, \mathrm{tr}\, (K_j K_i^{-1}) - n \tag{1.4}$$

where n is the order of the K_i's. Now since the K_i's are symmetric, positive-definite, we may write (modified Cholesky decomposition).

$$K_i = L_i G_i L_i^*$$

where L_i is unit lower triangular and G_i is diagonal. So we can write

$$\begin{aligned} tr\left[K_i K_j^{-1} \right] &= tr\left[L_i G_i L_i^* L_j^{*-1} G_j^{-1} L_j^{-1} \right] \\ &= tr\left[L_j^{-1} L_i G_i L_i^* L_j^{*-1} G_j^{-1} \right] \\ &= tr\left[T_{ji} G_i T_{ji}^* G_j^{-1} \right] \qquad (1.5) \end{aligned}$$

where

$T_{ji} = L_j^{-1} L_i$ and also is unit lower triangular. So (1.5) and a similar expression may be used to calculate D_1, using eq. (1.4) Now for D_2, we may rewrite (1.3) as

$$D_2 = \tfrac{1}{2} \left[(u_i - u_j)^* \; (K_i^{-1} + K_j^{-1}) \; (\mu_i - \mu_j) \right] \qquad (1.6)$$

Again, then we may use

$$K_i = L_i G_i L_i^*$$

and define

$$m_{ij} = \mu_i - \mu_j$$

So (1.6) may be rewritten as

$$\begin{aligned} D_2 &= \tfrac{1}{2} \left[m_{ij}^* \, (L_i^{*-1} G_i^{-1} L_i^{-1} + L_j^{*-1} G_j^{-1} L_j^{-1}) \, m_{ij} \right] \\ &= \tfrac{1}{2} \left[m_{ij}^* L_i^{*-1} G_i^{-1} L_i^{-1} m_{ij} + m_{ij}^* L_j^{*-1} G_j^{-1} L_j^{-1} m_{ij} \right] \\ &= \tfrac{1}{2} \left[x_i^* G_i^{-1} x_i + x_j^* G_j^{-1} x_j \right] \qquad (1.7) \end{aligned}$$

where

$$x_i = L_i^{-1} m_{ij}$$

and

$$x_j = L_j^{-1} m_{ij}$$

II. Computati n of the D(i,j)

To calculate the interclass divergence $D(i,j)$, first calculate the modified Cholesky decomposition of the two covariance matrices K_i and K_j

$$K_i = L_i G_i L_i^*$$

$$K_j = L_j G_j L_j^*$$

where L_i and L_j are unit lower triangular matrices and G_i and G_j are diagonal matrices. Using the notation $K_i = \left\{ k^i_{rs} \right\}$, $L_i = \left\{ \ell^i_{rs} \right\}$, and $G_i = \left\{ g^i_r \right\}$ we can write

$$\ell^i_{ss} = 1 \qquad\qquad s = 1, 2, \dots n \qquad\qquad (2.1)$$

$$g^i_1 = k^i_{11}$$

$$\left.\begin{aligned} g^i_s &= k^i_{ss} - \sum_{p=1}^{s-1} g^i_p \left(\ell^i_{sp} \right)^2 \\ \ell^i_{rs} &= \left(k^i_{rs} - \sum_{p=1}^{s-1} g^i_p \, \ell^i_{rp} \, \ell^i_{sp} \right) \Big/ g^i_s \end{aligned}\right\} s = 1, 2, \dots n$$

$$r = s+1 , s+2 , \dots n$$

with $\ell^i_{rs} = 0$ for $s > r$. Similar expressions hold for the elements of L_j and G_j. Next we need the elements of T_{ji} and T_{ij} $\left\{ t^{ji}_{rs} \right\}$ and $\left\{ t^{ij}_{rs} \right\}$

$$\left.\begin{aligned} t^{ji}_{rr} &= 1 \\ t^{ji}_{r,\,r-1} &= \ell^{i}_{r,\,r-1} - \ell^{j}_{r,\,r-1} \\ t^{ji}_{rs} &= \ell^{i}_{rs} - \ell^{j}_{rs} - \sum_{p=s+1}^{r-1} \ell^{j}_{rp}\, t^{ji}_{ps} \\ & s = r-2,\ r-3,\ \dots\ 1 \end{aligned}\right\}\ r=1,\ 2,\ \dots\ n \qquad (2.\,2)$$

and $t^{ji}_{rs} = 0$ for $s > r$. Similar expressions hold for t^{ij}_{rs}

Now form

$$m_{ij} = \mu_i - \mu_j \qquad (2.\,3)$$

and

$$\left.\begin{aligned} x^{i}_{r} &= m^{ij}_{r} - \sum_{p=1}^{r-1} \ell^{i}_{rp}\, x^{i}_{p} \\ x^{j}_{r} &= m^{ij}_{r} - \sum_{p=1}^{r-1} \ell^{i}_{rp}\, x^{i}_{p} \end{aligned}\right\}\quad r=1,\ 2,\ \dots\ n \qquad (2.\,4)$$

where $x_i = \left\{ x^{i}_{r} \right\}$, $x_j = \left\{ x^{j}_{r} \right\}$

Now $D(i,j)$ is given by

$$D(i,j) = \frac{1}{2} \sum_{r=1}^{n} \left\{ \sum_{\ell=1}^{r} \left[\left(t^{ji}_{r\ell} \right)^2 g^{i}_{\ell} / g^{j}_{r} + \left(t^{ij}_{r\ell} \right)^2 g^{j}_{\ell} / g^{i}_{r} \right] + \left(x^{i}_{r} \right)^2 / g^{i}_{r} + \left(x^{j}_{r} \right)^2 / g^{j}_{r} \right\} - n \qquad (2.\,5)$$

So the steps to form $D(i,j)$ are

1. Form L_i, G_i, L_j, and G_j using eqs. (2. 1)

2. Form T_{ji} and T_{ij} using eqs. (2.2)

3. Form m_{ij} using eq. (2.3)

4. Form x_i and x_j using eqs. (2.4)

5. Calculate $D(i, j)$ from eq. (2.5)

RICE UNIVERSITY

Institute for Computer Services
and Applications

MEMORANDUM

DATE: September 21, 1973

TO: K. Baker

FROM: D. L. Van Rooy

RE: Use of the Cholesky decomposition in D. Tebbe's feature selection

Analysis

Tebbe's method of feature selection consists of using a without replacement procedure for picking features, and classifying training fields to "determine" the probability of correct classification. At first, this may seem to be a quite time-consuming method. However, two points serve to expedite the procedure,

(1) the without replacement procedure greatly reduces the number of feature combinations to be used

and

(2) by partitioning the covariance matrices and judiciously saving appropriate results, the amount of computation may be greatly reduced both in computing the inverses of these matrices and classifying the training elements.

In his example, computation time for this method has been comparable to the exhaustive search, without-replacement divergence calculations, while classification accuracies have been greater than or equal to those using this divergence computation results.

The purpose of this note is to show how the execution time of Tebbe's method may be significantly decreased by employing the modified Cholesky decomposition$^{(2)}$. We have

$$K = LDL^*$$

where K is the covariance matrix, L is unit lower triangular (i.e. $\ell_{ii} = 1$, $\ell_{ij} = 0$ for $j > i$), D is a diagonal matrix and $*$ denotes transpose. The logarithm of the density function of a class (within a constant) is

$$f(x) = \tfrac{1}{2} \ell n \left|K\right| + \tfrac{1}{2}(x - \mu)^* K^{-1} (x-\mu) \qquad (1)$$

where u is the corresponding mean vector. So there are two problems to be attacked:

(i) how to economically obtain L_n and D_n given L_{n-1} and D_{n-1} where the subscript denotes the order of the matrix and

$$K_j = \left[\begin{array}{c|c} \boxed{K_{j-1}} & \nu \\ \hline \nu^* & \sigma \end{array}\right]$$

i.e. K_{j-1} is a submatrix of K_j (because of the without replacement procedure), ν is a vector, and σ a scalar.

and

(ii) given L and D how does one economically evaluate eq. (1).

Now it is easy to show that

$$\left.\begin{aligned} \ell_{ni}^{(n)} &= \left(k_{ni}^{(n)} - \sum_{j=1}^{i-1} \ell_{ij}^{(n-1)} d_j^{(n-1)} \ell_{nj}^{(n)}\right) \Big/ d_i && (2a) \\ \ell_{ij}^{(n)} &= \ell_{ij}^{(n-1)} \qquad j = 1, 2, \ldots i && (2b) \end{aligned}\right\} i=1, 2, \ldots n-1$$

where $L_n = \left\{ \ell_{ij}^{(n)} \right\}$ and similarly for K and D.

also

$$d_n^{(n)} = k_{nn}^{(n)} - \sum_{p=1}^{n-1} \ell_{np}^{(n)^2} d_p^{(n-1)} \qquad (3a)$$

and

$$d_i^{(n)} = d_i^{(n-1)} \qquad\qquad i = 1,\ 2,\ \ldots\ n-1 \qquad (3b)$$

Thus L_n differs from L_{n-1} only in the last row, and D_n differs from D_{n-1} only in the nn^{th} position.

We note that

$$\begin{aligned} \left|K_n\right| &= \left|L_n\ D_n\ L_n^*\right| \\ &= \left|L_n\right|\ \left|D_n\right|\ \left|L_n^*\right| \\ &= \left|D_n\right| \end{aligned}$$

$$\left|K_n\right| = d_n^{(n)} \cdot \left|D_{n-1}\right| \qquad (4)$$

Rewriting eq. (1) we have

$$f(x) = \tfrac{1}{2} \ln \left|K_{n-1}\right| + \tfrac{1}{2} \ln d_n^{(n)} + \tfrac{1}{2}\, y^*\, L_n^{*-1}\ D_n^{-1}\ L_n^{-1}\ y \qquad (5)$$

where $y = x - \mu$

Defining $z = L_n^{-1}\ y$

$$= \begin{pmatrix} z^{(1)} \\ z^{(2)} \end{pmatrix} \quad \text{with } z^{(2)} \text{ a scalar}$$

we obtain for the last term in (5)

$$z^*\ D_n^{-1}\ z$$

$$= z^{(1)*}\ D_{n-1}^{-1}\ z^{(1)} + \left(z^{(2)}\right)^2 d_n^{(n)}$$

So (5) becomes

$$f(x) = \tfrac{1}{2}\left(\ln \left|K_{n-1}\right| + z^{(1)*}\, D^{-1}_{n-1}\, z^{(1)} \right) + \tfrac{1}{2}\left(\ln d_n^{(n)} + \left(z^{(2)}\right)^2 \Big/ d_n^{(n)} \right) \qquad (6)$$

with

$$z_i = y_i - \sum_{j=1}^{i-1} \ell_{ij}^{(n-1)}\, z_j \qquad i = 1, 2, \ldots n-1 \qquad (7a)$$

$$z_n = y_n - \sum_{j=1}^{n-1} \ell_{nj}^{(n)}\, z_j \equiv z^{(2)} \qquad (7b)$$

So we wee that the first two terms in eq. (6) do not depend on any values from the n^th channel. Thus they may be precomputed and used with each of the channels.

Comparing these results with Tebbe's, we note that the classification of each point will require the same amount of computation after the leading term (his $\Delta_o' \sum_o^{-1} \Delta_o$) has been computed. However, the proposed method is faster and more numerically stable since

1) no matrix inverses need be computed, and

2) the calculation of f in Tebbe's method requires $\sim n^2$ multiplications whereas the corresponding calculation of $\ell^{(n)}$ requires $\sim \frac{n^2}{2}$ multiplications.

Implementation

The above results can be readily utilized in the existing program. The following describes the changes necessary (see ref. (1)) :

1) Tebbe's $\sum_o^{-1}$ is not computed. Instead, the modified Cholesky decomposition (= L DL *) is computed (see ref (2))(it could be saved from the prior case where the best n-1 channels were computed). This yields the $\ell_{ij}^{(n-1)}$ and $d_j^{(n-1)}$ used in eqs. (2a) & (2b).

2) In place of the calculations of his e and f will be $\ell_{ni}^{(n)}$ i = 1, 2, . . . n-1 and $d_n^{(n)}$ from eqs. (2a) & (3a) in this report. (Note that the product $\ell_{ij}^{(n-1)}$ $d_j^{(n-1)}$ in eq. 2a may be precomputed).

3) His ℓn e becomes $\ell n\ d_n^{(n)}$

4) His S_o is now $\ell n \left|K_{n-1}\right| + z^{(1)*} D_{n-1}^{-1} z^{(1)}$

where $\left|K_{n-1}\right| = \prod_{i=1}^{n-1} d_i^{(n-1)}$ and $z^{(1)*} D_{n-1}^{-1} z^{(1)} = \sum_{i=1}^{n-1} z_i^2 / d_i$

and z_i is given by eq. (7a)

5) The term $\frac{1}{e}\ (f\ '\ \Delta_o - \delta)^2$ in S in his report becomes $\left(z^{(2)}\right)^2 \Big/ d_n^{(n)}$

where $z^{(2)}$ is given by eq. (7b).

References

1. Dennis L. Tebbe, "Feature Selection Software for Pattern Recognition applied to Multispectral Earth Resources Data," NASA - EOD internal memo, August 10, 1973.

2. D. L. Van Rooy, M. S. Lynn, and C. H. Snyder, "The Use of the Modified Cholesky Decomposition in Divergence and Classification Calculations," ICSA Technical Report #275-025-008, Rice University, Houston, Texas, May, 1973.

RICE UNIVERSITY

Institute for Computer Services
and Applications

MEMORANDUM

DATE: November 30, 1973

TO: Ken Baker

FROM: D. L. Van Rooy

RE: Improved method for computing the average weighted divergence

The average weighted divergence may be written, following Quirein(1), as:

$$D = \frac{1}{2} \sum_{i=1}^{m} \mathrm{tr}\left\{ K_i^{-1} \; S_i \right\} - n \sum_{i=1}^{m-1} \; \sum_{j=i+1}^{m} W_{ij} \qquad (1)$$

where K_i is the i^{th} covariance matrix

m is the number of classes

n is the dimensionality

and

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^{m} W_{ij} \; (K_j + \delta_{ij} \; \delta_{ij}^{*}) \qquad (2)$$

with W_{ij} being the weighting factors

and δ_{ij} being the difference of the means between classes i and j

Now we note that both the K_i and S_i are symmetric, positive-definite matrices. Thus we may write

$$K_i = L_i G_i L_i^* \quad \text{(modified Cholesky decomposition)} \tag{3a}$$

and

$$S_i = R_i + R_i^* \tag{3b}$$

where L_i and R_i are lower triangular and G_i a diagonal matrix. So the term $K_i^{-1} S_i$ in eq. (1) becomes

$$K_i^{-1} S_i = L_i^{*-1} G_i^{-1} L_i^{-1} (R_i + R_i^*)$$

and

$$\begin{aligned} \mathrm{tr}(K_i^{-1} S_i) &= \mathrm{tr}(G_i^{-1} L_i^{-1} R_i L_i^{*-1} + L_i^{-1} R_i^* L_i^{*-1} G_i^{-1}) \\ &= \mathrm{tr}(Q_i + Q_i^*) \\ &= 2\,\mathrm{tr}(Q_i) \end{aligned}$$

where

$$Q_i = G_i^{-1} L_i^{-1} R_i L_i^{*-1} \tag{4}$$

So now eq. (1) becomes

$$D = \sum_{i=1}^{m} \mathrm{tr}(Q_i) - n \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} W_{ij} \tag{5}$$

We note that only the diagonal elements of Q_i are needed,

$$q^i_{jj} = \left(L_i^{-1} \; R_i \; L_i^{*-1} \right)_{jj} \Big/ g^i_j \qquad (6)$$

where $Q_i = \left(q^i_{ju} \right)$

$$G_i = \left(g^i_j \right)$$

$$L_i = \left(\ell^i_{ju} \right) \qquad \text{with} \quad \ell^i_{jj} = 1$$

$$R_i = \left(r^i_{ju} \right)$$

$$S_i = \left(s^i_{ju} \right)$$

$$K^i = \left(k^i_{ju} \right)$$

First form

$T_i^* = L_i^{-1} \; R_i$ where T_i is upper triangular

which requires $\dfrac{n\,(n+1)\;(n+2)}{6}$ multiplies

Then compute the diagonal elements of $L_i^{-1}\, T_i = C_i$. This also $\dfrac{n\,(n+1)\,(n+2)}{6}$ multiplies. So eq. (6) becomes

$$q^i_{jj} = c^i_{jj} \Big/ g^i_j$$

Thus the total number of multiplies involved is

$$\frac{2n\ (n+1)\ (n+2)}{3} + n \qquad \text{which is} \quad O\left(\frac{2n^3}{3}\right),$$

including the multiplications necessary to perform the modified Cholesky decomposition.

Algorithm

1). Compute the S_i's according to eq. (2) using all channels. Also compute the last term in eq. (5) .

2). Form the R_i's using eq. (3b) .

$$\text{i.e.} \quad \left.\begin{aligned} r^i_{jk} &= s^i_{jk} \\ r^i_{jj} &= \tfrac{1}{2}\, s^i_{jj} \\ r^i_{kj} &= 0 \end{aligned}\right\} \qquad j > k$$

3). For particular conbinations of channels,pick out the submatrices K^i and R^i

4). Form L_i and G_i as in eq. (3a).

$$g^i_1 = k^i_{11}$$

$$\left.\begin{aligned}
g^i_j &= k^i_{jj} - \sum_{u=1}^{j-1} g^i_u \left(\ell^i_{ju}\right)^2 \\
\ell^i_{vj} &= \left(k^i_{vj} - \sum_{u=1}^{j-1} g^i_u \ \ell^i_{vu} \ \ell^i_{ju}\right) \Big/ g^i_j \\
& \qquad v = j+1,\ j+2,\ \ldots\ n
\end{aligned}\right\} \quad j = 1,\ 2,\ \ldots\ n$$

with $\ell^i_{jj} = 1$ and $\ell^i_{vj} = 0$ for $j > v$

5). Compute the upper triangular elements of T_i using

$$t^i_{jj} = r^i_{jj} \qquad j = 1,\ 2,\ \ldots\ n$$

$$\left.\begin{aligned}
t^i_{jk} &= r^i_{kj} - \sum_{u=1}^{k-1} \ell^i_{ku} \ t^i_{uj} \\
& \qquad j = k+1,\ k+2,\ \ldots\ n
\end{aligned}\right\} \quad k = 1,\ 2,\ \ldots\ n-1$$

6). Compute the following elements of C_i.

$$\left.\begin{aligned}
c^i_{jk} &= t^i_{jk} - \sum_{u=1}^{j-1} \ell^i_{ju} \ c^i_{uk} \\
& \qquad j = 1,\ 2,\ \ldots\ k
\end{aligned}\right\} \quad k = 1,\ 2,\ \ldots\ n$$

[N.B. only the diagonal elements of C_i are needed but the others are necessary for the calculation of these elements.]

7). Form

$$D = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{jj}^{i} \Big/ g_{j}^{i} - n \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} W_{ij} ,$$

the average weighted divergence.

Reference

1) John Quirein, "Divergence: Some Necessary Conditions for Extremum," U. of Houston Technical Report, Mathematics Dept., November, 1972.